

CLAIMS

What is claimed is:

1. A method for providing efficient bi-directional communications between a client computer and at least one server computer, the at least one server computer hosting at least one message queuing middleware system, the method comprising the steps of:

providing at least one client computer code module resident on the client computer, wherein the step of providing the at least one client computer code module further comprises the steps of:

providing at least one client code adapter on the client computer, wherein the client code adapter comprises:

providing a first module adaptable to remote message queuing service;

providing a second module for associating a data message with the least one message queuing middleware system;

transmitting the associated data message to the at least one server computer;

receiving the associated data message on the at least one server computer;

providing a server computer code module resident on the at least one server computer, wherein the step

of providing the server computer code module further comprises the steps of:

configuring the server computer code module to select at least one messaging middleware protocol set based upon information contained within the received associated data message;

translating the associated data message to conform to the selected messaging middleware protocol set; and

returning a status message to the client computer.

2. A method as in claim 1 wherein the step of transmitting the associated data message to the at least one server further comprises the step of transmitting the associated data message according to a communication protocol.

3. A method as in claim 2 wherein the step of transmitting the associated data message according to the communication protocol further comprises the step of transmitting the associated data message according to the Transmission Control Protocol/Internet Protocol (TCP/IP).

4. A method as in claim 2 wherein the step of the step of transmitting the associated data message according to the communication protocol further comprises the step of transmitting the associated data message according to a wireless communication protocol.

5. A method as in claim 1 wherein the step of providing at least one client computer code module resident on the client computer further comprises the step of providing the at least one client computer code module to be compatible with JAVA language programming code.

6. A method as in claim 1 wherein the step of configuring the server computer code module to select at least one messaging middleware protocol set further comprises the step of:

selecting the at least one messaging middleware protocol set from the group of messaging middleware sets consisting of a first messaging protocol, a second messaging protocol, and a third messaging protocol.

7. A system for performing efficient web based proxy messaging for message queuing middleware, the system comprising:

a client computer, wherein the client computer comprises:

at least one user application;

at least one client software module, wherein the at least one client software module comprises:

a data connection to the at least one user application;

a selector for associating a data message with at least one message queuing middleware system;

a transmitter for transmitting the data message via a computer network;

at least one server computer having an interface for communicating via the computer network, the server computer comprising:

a translator; and

at least one message queuing middleware system connectable to the translator.

8. A system as in claim 7 wherein at least one client software module comprises:

at least one library, wherein the at least one library comprise software code adaptable to compiling.

9. A system as in claim 8 wherein the software code is C language code.

10. A system as in claim 7 wherein the at least one client software module comprises:

at least one class, wherein the at least one class is adaptable to interfacing with JAVA programming code.

11. A system as in claim 7 wherein the transmitter further comprises the transmitter adaptable to transmitting the associated data message according to a communication protocol.

12. A system as in claim 11 wherein the transmitter for transmitting the associated data message according to a communication protocol further comprises the transmitter adaptable to transmitting the associated data message according to the Transmission Control Protocol/Internet Protocol (TCP/IP).

13. A system as in claim 11 wherein the transmitter for transmitting the associated data message according to a communication protocol further comprises the transmitter adaptable for transmitting the associated data message according to a wireless protocol.

14. A system as in claim 7 wherein the translator further comprises the translator adaptable for selecting at least one messaging middleware protocol set.

15. A system as in claim 7 wherein the selector for selecting the at least one messaging middleware protocol set further comprises the selector adaptable for selecting the at least one messaging middleware protocol set from the group of messaging middleware sets consisting of a first messaging protocol, a second messaging protocol, and a third messaging protocol.

16. A method for providing an efficient proxy message method for message queuing middleware hosted on at least one server computer having access to at least one client computer, the method comprising the steps of:

receiving at least one function call from the client computer;

translating the at least one function call to at least one receiver format;

transmitting the at least one function call translated into at least one receiver format to the at least one server; and

executing the at least one function call translated into at least one receiver format.

17. A method as in claim 16 wherein the step of receiving the at least one function call from the client computer further comprises the step of applying the function call to a client software module.

18. A method as in claim 16 wherein the step of translating the at least one function call to at least one receiver format further comprises the step of translating the function call into a predetermined message queuing protocol.

19. A method as in claim 16 wherein the step of transmitting the at least one function call translated into at least one receiver format further comprises the step of transmitting via a communications protocol.

20. a method as in claim 16 wherein the step of executing the at least one function call translated into

at least one receiver format further comprises the step of returning a message resulting from the executed function call.

21. At least one program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for providing efficient bi-directional communications between a client computer and at least one server computer, the method comprising the steps of:

providing at least one client computer code module resident on the client computer, wherein the step of providing the at least one client computer code further comprises the steps of:

providing at least one library resident on the client computer, wherein the library comprises:

means for remote message queuing service;

means for associating a data message with the least one message queuing middleware system;

transmitting the associated data message to the at least one server computer;

receiving the associated data message on the at least one server computer;

providing a server computer code module resident on the at least one server computer, wherein the step

of providing the server computer code module further comprises the steps of:

configuring the server computer code module to select at least one messaging middleware protocol set based upon information contained within the received associated data message;

translating the associated data message to conform to the selected messaging middleware protocol set; and

returning a status message to the client computer.

2025-04-04 10:00:00